

AD-A166 009 EQUATIONAL PROGRAMMING (A STUDY IN EXECUTABLE ALGEBRAIC 1/1
FORMAL SPECIFICATION)(U) SRI INTERNATIONAL MENLO PARK
CA COMPUTER SCIENCE LAB J A GOGUEN ET AL. 05 MAR 86
UNCLASSIFIED N00014-82-C-0333 F/G 9/2 NL



AD-A166 009

SRI International



DTIC FILE COPY

EQUATIONAL PROGRAMMING
(A Study in Executable Algebraic Formal Specification)

Final Report

5 March 1986

By: Joseph A. Goguen, Program Manager
José Meseguer, Sr. Computer Scientist

Computer Science Laboratory
Computer Science and Technology Division

SRI International

Prepared for:
Office of Naval Research
800 Quincy Street
Arlington, Virginia 22217

Attention: Dr. Charles Holland

Contract Number N00014-82-C-0333

SRI Project ECU 4406

DISTRIBUTION STATEMENT A
Approved for public release
Distribution Unlimited

Approved:

Donald L. Nielson, Vice President and Director
Computer Science and Technology Division

333 Ravenswood Ave • Menlo Park, CA 94025
415 326-6200 • TWX 910-373-2046 • Telex 334 486

86 3 10 057

12

DTIC
ELECTE
APR 03 1986
S D

This report summarizes progress on the Equational Programming project sponsored by the office of Naval Research, during the period July 1, 1982 to December 31, 1985. Most of our effort during this period was devoted to designing and beginning to implement the OBJ2 equational programming language. This language was designed and implemented in accord with the rigorous standards of logic and mathematics; we believe this to be the essence of logic programming. Moreover, a rigorous semantics leads to considerable savings in implementation effort, and need not restrict expressive power. Thus, the work on implementing OBJ2 has been closely coupled with work on its semantics, and the implementation has made essential use of various conceptual advances.

The overall conception of OBJ2, with its new ideas on inheritance, views, and parameterization, was sketched by Goguen [5]. A semantic integration of Horn clause (logic) and equational (functional) programming was done by Goguen and Meseguer in their design and semantics for the Eqlog language [13], containing OBJ2 as its functional sublanguage. Inheritance based upon subsorts is fundamental to both OBJ2 and Eqlog and has involved advances in both mathematical and operational semantics [14, 4, 17].

The potential of algebraic techniques in computer science has been limited in part by the awkwardness of handling partially defined operations and exceptions algebraically. We have developed new techniques for these and related problems (such as polymorphism, inheritance, and objects with local states) that are as simple and powerful as the familiar total universal algebra, but are much more flexible and expressive. A major advance has been to extend the fundamental results of abstract data types to order-sorted algebra (OSA), so as to support inheritance, polymorphism, error handling, and partiality [14]. This extension includes initial and free algebras, parameterization, completeness of equational deduction, and partial algebras with equationally defined domains. OSA provides a rigorous mathematical basis for OBJ2, and all the power of OSA is directly reflected and supported by the OBJ2 implementation [4].

or
 &l
 d

lts on file

ility Codes

il and/or
 Special

Dist
 A-1

Another advance has been the development of an efficient operational semantics for OSA, based upon rewrite rules [17]. By reducing the problem to standard term rewriting in ordinary equational logic, we have achieved a very expressive language implemented at the less expressive but more efficient level of standard equational logic [23]. In this way, not only can well-understood techniques and algorithms be used for implementation, but also the many theorem proving tools and algorithms already developed for equational logic can be used to prove properties of OBJ2 programs [20].

The following is a more detailed enumeration of what has been accomplished:

1. In collaboration with David Plaisted, starting from the OBJT system [16], we developed OBJ1. This demonstrated the viability of the basic ideas of OBJ2, including generic objects, subsorts, interactive mixfix syntax, user-definable abstract data types, and implementing equational logic with rewrite rules. The limitations of the way some of these features were implemented led us to the much more ambitious OBJ2 system.
2. Goguen sketched OBJ2 in "Parameterized Programming" [5]. Although considerable work on the OBJ2 semantics and implementation remained at that point, the essential features of OBJ2 are as outlined there.
3. Goguen and Meseguer have developed an equational logic programming language called OBJ2 that is based on OSA, in joint work with Kokichi Futatsugi and Jean-Piere Jouannaud [4]. This language has many novel features, including parameterization (after the style of the specification language Clear [2, 3]) with requirement theories for module interfaces and theory morphisms ("views") for binding actuals to formals, multiple inheritance, runtime typechecking by "retracts," and user-definable built-ins that can be implemented in the underlying LISP, all with a rigorous logical semantics.
4. Goguen showed how the ideas of OBJ2 could be used as a rigorous high-level module interconnection language for Ada [6, 10], thus supporting

programming in the large and providing a good basis for Ada libraries and environments.

5. Goguen and Meseguer have developed a generalization of many-sorted universal algebra called "Order-Sorted Algebra" (OSA) which extends the algebraic approach to abstract data types (ADTs) to handle both parametric and "ad hoc" polymorphism (in the sense of Strachey [24]), multiple inheritance, error specification and recovery, and partial functions [14]. This removes some long-standing obstacles to the wider practical and theoretical application of ADTs. The main results of universal algebra extend to OSA, including existence of initial algebras, completeness of equational deduction, Birkhoff variety and Malcev quasivariety theorems [14]. Moreover, a rewrite-rule based operational semantics for OSA has been developed jointly with Jean-Pierre Jouannaud, having the same nice properties as the one-sorted case [17]. This provides a rigorous mathematical and operational semantics for the OBJ2 language.
6. Goguen and Futatsugi have developed a "Command Interpreter Generator" that will automatically generate a menu-driven user interface for interactive systems like OBJ2. This is, in itself, a very powerful software development tool that has excited considerable interest (especially abroad), since one can easily add language features (or modify them) without having to recode a very complex user interface.
7. Goguen and Meseguer have developed an algebraic theory of software modules (i.e., "persistent objects" having states) that is a fairly straightforward generalization of the algebraic theory of abstract data types. This theory uses "hidden sorts" to account for internal states and shows that many of the concepts of classical automaton theory extend to this area of "abstract software machines" [12]. This will provide a basis for extending OBJ to encompass object-oriented programming.
8. Goguen and Meseguer have shown that the standard rules of one-sorted equational deduction are not sound for many-sorted equational logic, and

have given a new set of rules that is both sound and complete [11]; prior to this, workers in abstract data types had used the one-sorted rules uncritically (e.g., Goguen, Thatcher, and Wagner [18]). In later work, Meseguer and Goguen showed that it is valid to use these one-sorted rules in some important special cases, including the case of term rewriting. This leads to the more efficient operational semantics used in OBJ2 [21].

9. Goguen and Meseguer have also designed a logic programming language called Eqlog that combines predicate-based logic programming *a la* Prolog with equational functional programming as in OBJ2, by unifying their underlying logics, as Horn clause logic with equality [13, 15]. Eqlog can be seen as a "constraint language" in which equations for built-ins and for user-defined ADTs can be solved by "narrowing." It will be implemented as an extension of OBJ2, and will include all the OBJ2 features mentioned above. These languages are implemented at many sites throughout the world, including the Universities of Milan, Manchester, New Hampshire and Oregon, and are being considered at others, including Tektronics, Olivetti, MCC and Hewlett-Packard.
10. Goguen has developed a very general theory called "Institutions" in joint work with Prof. Rodney Burstall [9, 8]. This provides an abstract model theory (in roughly the sense of Barwise [1]) that can relate the many different logics used in computer science; it also gives a general method for providing parameterized modules in any pure logic programming language. Different logics can be related by "institution morphisms," providing a rigorous link between different theorem provers and specification languages, as well as a rigorous basis for integrating multiple underlying logics in program design, specification, verification, and in logic programming.
11. Meseguer and Goguen have written a comprehensive survey of the theory of abstract data types, also including some new results and ideas [22]. This survey discusses the following topics in some depth: the importance of abstract data types in software methodology, equational deduction, rewrite

rules; abstract machines (i.e., ADTs with hidden sorts), and computability theory for ADTs.

12. Meseguer, in joint work with Irene Guessarian, has given a complete equational axiomatization of if-then-else for proving properties about functional programs over user-definable ADTs [19]. This will support proof systems based on the algebraic paradigm of using equational methods for program semantics and verification.
13. In very recent work, Meseguer and Goguen have proven that *any* partial recursive function can be specified equationally. This proves that an equational programming language like OBJ2 has the full power of recursive function theory, and settles a long-standing problem concerning the power of algebraic specifications that do not necessarily terminate as rewrite rules.
14. In recent work with Lawrence Moss, Goguen and Meseguer have also solved an important outstanding conjecture on the computational power of final algebra semantics. Their result shows that any cosemicomputable data type with nontrivial behavior can be axiomatized by final algebra semantics using a finite set of equations. They also show that all the recursive enumerable degrees of unsolvability can be realized by final algebra semantics, and they characterize the finite signatures of function symbols such that all the degrees of unsolvability can be realized by quotients of its initial algebra.

References

- [1] Barwise, Jon.
Axioms for Abstract Model Theory.
Annals of Mathematical Logic 7:221-265, 1974.
- [2] Burstall, R. M. and Joseph Goguen.
Putting Theories together to Make Specifications.
Proceedings, Fifth International Joint Conference on Artificial Intelligence :1045-1058, 1977.
- [3] Burstall, R. M. and Joseph Goguen.
The Semantics of Clear, a Specification Language.
In *Lecture Notes in Computer Science*. Volume 86: *Proceedings of the 1979 Copenhagen Winter School on Abstract Software Specification*, pages 292-332. Springer-Verlag, 1980.
- [4] Futatsugi, Kokichi, Joseph Goguen, Jean-Pierre Jouannaud and José Meseguer.
Principles of OBJ2.
In *Proceedings, Symposium on Principles of Programming Languages*, pages 52-66. Association for Computing Machinery, 1985.
- [5] Goguen, Joseph.
Parameterized Programming.
Transactions on Software Engineering SE-10(5):528-543, September, 1984.
- [6] Goguen, Joseph.
Suggestions for Using and Organizing Libraries in Software Development.
In Steven Kartashev and Svetlana Kartashev (editor), *Proceedings, First International Conference on Supercomputing Systems*, pages 349-360. IEEE Computer Society Press, 1985.
Preliminary version in [10]; final version in [7].
- [7] Goguen, Joseph.
Reusing and Interconnecting Software Components.
IEEE Computer 19(2):16-28, February, 1986.
- [8] Goguen, Joseph and R. M. Burstall.
Introducing Institutions.
In Edward Clarke and Dexter Kozen (editor), *Lecture Notes in Computer Science*. Volume 164: *Proceedings, Logics of Programming Workshop*, pages 221-256. Springer-Verlag, 1984.

- [9] Goguen, Joseph and R. M. Burstall.
Institutions: Abstract Model Theory for Computer Science.
 Technical Report CSLI-85-30, Center for the Study of Language and
 Information, Stanford University, 1985.
 Also submitted for publication; a preliminary version appears in
Proceedings, Logics of Programming Workshop, edited by Edward
 Clarke and Dexter Kozen, volume 164, Springer-Verlag Lecture Notes
 in Computer Science, pages 221-256, 1984.
- [10] Goguen, Joseph and Karl Levitt.
Report on Ada Program Libraries Workshop.
 Technical Report, SRI International, Computer Science Lab, 1984.
 Held at Naval Postgraduate School, Monterey CA, November 1-3, 1984;
 Prepared for Ada Joint Program Office.
- [11] Goguen, Joseph and José Meseguer.
 Completeness of Many-sorted Equational Logic.
SIGPLAN Notices 16(7):24-32, July, 1981.
 Also appeared in *SIGPLAN Notices*, January 1982, vol. 17, no. 1, pages
 9-17; extended version as SRI Technical Report CSL-135, May 1982;
 final version as Technical Report CSLI-84-15, Center for the Study of
 Language and Information, Stanford University, September 1984, and
Houston Journal of Mathematics, volume 11, number 3, pages
 307-334, 1985.
- [12] Goguen, Joseph and José Meseguer.
 Universal Realization, Persistent Interconnection and Implementation of
 Abstract Modules.
 In *Lecture Notes in Computer Science. Volume 134: Proceedings, 9th
 International Conference on Automata, Languages and Programming*,
 pages 265-281. Springer-Verlag, 1982.
- [13] Goguen, Joseph and José Meseguer.
 Equality, Types, Modules and (Why Not?) Generics for Logic
 Programming.
The Journal of Logic Programming 1(2):179-210, September, 1984.
- [14] Goguen, Joseph and José Meseguer.
*Order-Sorted Algebra I: Partial and Overloaded Operations, Errors and
 Inheritance.*
 Technical Report to appear, SRI International, Computer Science Lab,
 1986.
 Given as lecture at Seminar on Types, Carnegie-Mellon University, June
 1983.

- [15] Goguen, Joseph and José Meseguer.
Eqlog: Equality, Types, and Generic Modules for Logic Programming.
In Douglas DeGroot and Gary Lindstrom (editors), *Functional and Logic Programming*. Prentice-Hall, 1986.
An earlier version appears in the *Journal of Logic Programming*, volume 1, number 2, pages 179-210, September 1984.
- [16] Goguen, Joseph and Joseph Tardo.
An Introduction to OBJ: A Language for Writing and Testing Software Specifications.
In *Specification of Reliable Software*, pages 170-189. IEEE Press, 1979.
reprinted in *Software Specification Techniques*, edited by N. Gehani and A. D. McGettrick, Addison-Wesley, 1985, pages 391-420.
- [17] Goguen, Joseph, Jean-Pierre Jouannaud and José Meseguer.
Operational Semantics of Order-Sorted Algebra.
In *Proceedings, 1985 International Conference on Automata, Languages and Programming*. Springer-Verlag, 1985.
Summary presented at IFIP WG2.2 (Boston MA) June 1984.
- [18] Goguen, Joseph, James Thatcher and Eric Wagner.
An Initial Algebra Approach to the Specification, Correctness and Implementation of Abstract Data Types.
Technical Report RC 6487, IBM T. J. Watson Research Center, October, 1976.
Reprinted in *Current Trends in Programming Methodology, IV*, edited by Raymond Yeh, Prentice-Hall, 1978, pages 80-149.
- [19] Guessarian, Irene and José Meseguer.
On the Axiomatization of 'if-then-else'.
Technical Report 20, Center for the Study of Language and Information, Stanford University, 1985.
- [20] Lescanne, Pierre.
Computer Experiments with the REVE Term Rewriting Systems Generator.
In *Proceedings, Symposium on Principles of Programming Languages*. Association for Computing Machinery, 1983.
- [21] Meseguer, José and Joseph Goguen.
Deduction with Many-Sorted Rewrite Rules.
Technical Report CSLI-85-42, Center for the Study of Language and Information, Stanford University, December, 1985.

- [22] Meseguer, José and Joseph Goguen.
Initiality, Induction and Computability.
In Maurice Nivat and John C. Reynolds (editors), *Algebraic Methods in Semantics*, pages 459-541. Cambridge University Press, 1985.
Chapter 14; also SRI CSL Technical Report 140, December 1983.
- [23] O'Donnell, Michael.
Equational Logic as a Programming Language.
MIT Press, 1985.
- [24] Strachey, Christopher.
Fundamental Concepts in Programming Languages.
Lecture Notes from International Summer School in Computer Programming, Copenhagen.
1967

END

DTIC

4-86